IoT Enabled Aquatic Drone for Environmental Monitoring

João Matos, Octavian Postolache
ISCTE-Instituto Universitário de Lisboa, ISCTE-IUL, Instituto de Telecomunicações,
Avenida das Forças Armadas, 1649-026 Lisboa
email: jrbms@iscte.pt; opostolache@lx.t.pt

Abstract—**The article describes a monitoring systembased on Raspberry Pi platform and a multichannel sensing module associated with water quality and air quality measurement parameters. Thus the temperature, conductivity, relative humidity and gas concentration are measured as was as the underwater acoustic signals. The data is stored on the memoryof the drone's computational platform, and synchronized with a remote server database. Advanced data processing algorithms were implemented on the server side. Additionally, a mobile application was developed to be used by people working in the field for data visualization and statistical analysis.**

Keywords—Drones; USV;Sensors; Hydrophone; Raspberry P; Streaming.

I. INTRODUCTION

The Internet of Things is an already known term nowadays and it is becoming bigger and bigger overtime with all sorts of sensors and systems being developed to help people ease up their lives. The number of connected devices continues to grow worldwide but also the diversity and the applications in the real world are immense, making it an appealing industry to work on [1].A lot of companies are working on new devices and new solutions to deal with the growth of the connected devices, it is a massive growing industry. It possesses the power to transform every single environment such as agriculture, transportation, manufacturing, smart houses even entire cities. Companies are working on making new devices for every possible scenario but also improving communication protocols as well as security. It is estimated that around 2015 10 billion devices were connected and by 2020 will be connected 20 to 30 billion devices, as costs continue to drop and demand continues to grow [2].

Rivers and estuaries are incredibly important, they are a source of fresh water and nutrients required by animals and human activity, but with the development of society and industrial demands, the pollution of rivers have become a problem to be concerned about. Food production and pollution have been putting stress within these natural resources and it is necessary to create systems to help maintain the good health of rivers. Other factor that should not be ignored is the noise caused by human activity around the rivers but also by the animals themselves, it is interesting to see the relation of these two and see if outside noise can disrupt natural activities from the rivers [3].

With today's advancements in technology, it became possible to address a larger number of problems regarding accessibility and also monitoring and interacting more thoroughly with systems. This workpresents the design and implementation of a system that can be used for water quality monitoring, the system being designed especially for monitoring tasks onrivers and estuaries.The capabilities of the system is based on the usage of unmanned surface vehicle (USV) that helpson maritime monitoring tasks using multi-drone swarms [4] for extended spatial resolution. The system may detect such as cleaning oil spill [5], or for general environmental condition monitoring.

The Internet of Things help to solve a lot of problems in different fields: in smart cities, IoT applications are related with parking issues, noise, traffic, illumination monitoring [6]; emergency systems for earthquakes [7]; precision agriculture applications in culture process optimization [8]. IoT are used to deliver information from the sensors and to the actuators.

The goal of the paper is to present the aquatic drone setup equipped with a Raspberry Pi that is connected to an array of sensors for air and the waterquality monitoring. For air quality measurements, temperature, humidity and gas sensors were considered. For water quality monitoring water temperature and conductivity sensing channels were implemented.

Several tasks were carried out during the system implementation. Thus the sensors interfacing with Raspberry Pi platform was considered, different signal conditioning modules were implemented. Referring software Python scripts were developed in order to read the values from the sensors and to write them to a local database. A server setup was carried out to receive the data from the local database implemented on Raspberry Pi level using database replication, in which the server database will always update whenever there are changes in the sensor readings. In order to be possible to access and visualize the data in an appropriate GUIanandroid application was developed. The applications fetch the data from the server database and provide live readings for the system user.

II. RELATED WORK

The BioMachines Lab in Institute of Telecommunications in ISCTE-IUL, Lisbon is developing a swarm of aquatic drones designed to perform maritime tasks in the sea based onaWi-Fi ad-hoc network architecture. Each drone is equipped with a compass, GPS and a Raspberry Pi 2 unit with a Wi-Fi interface and they form a distributed network (controlled by an artificial neural network based controller) without a central coordination, they keep broadcasting their position to the drones in their neighborhood every second. The neural network performs actions depending on what it receives, the sensor

978-1-5090-6129-7/16/$31.00 ©2016 European Union

readings feed the neural network and then it will control the drone based on the information received [9].

Reported research at the International Conference on Circuit, Power and Computing Technologies (ICCPCT) 2015 consisted ofdeveloping of a static system that includes Raspberry Pi B+ model connected to an extra sensorssuch as: temperature sensor. The Raspberry is part of anIoT module that will send the data to a cloud server, the cloud server then sends the data to a web server with a user interface [10].

In the United States, a project called the Water on the Web (WOW) gathers information from lakes and rivers across the different states of the country, it uses a RUSS (Remote Underwater Sampling Station) which is a floating platform equipped with conductivity, dissolved oxygen, pH, temperature and turbidity sensors and it uses cellular networks to transmit the data to the website [11].

This project resembles in a way to all of these three projects and will sort of connect them together, it is not as thorough as some of them but targets the main subjects.

Drones have come a long way and have all sorts of applications and come in different shapes, UAVs (aerial), USVs (water surface), UUVs (underwater) and UGVs (on the ground) [12][13]. The UAVs are perhaps the most common with their vast application in military and the continuous investments in that way [14]. Nowadays the drone market is targeting more domestic and commercial usage like helping weather forecasting [15], agriculture, movie industry, real estate, local law enforcements, the construction business [16], and much more that can be explored. All of this presents a new large potential to dramatically change people's lives, whole industries and also introducing more privacy concerns and security [17].

III. System Description

The developed system based on Raspberry pi platform is characterized by a hardware and software parts. The hardware part of the system is represented in the Fig. 1:

The analog sensors are connected to the Raspberry Pi to an ADC board based on ADS1015. The hydrophone channel requires special analog to digital conversion and interfacing capabilities thus an USB Audio Card was employed and connected to a Raspberry USB port.

The software component part and the hardware software relations are presented in Fig. 2.

As it is presented in the Fig. 2 the Raspberry has a local database that is fetching the data from the sensors, expressed by water and air measured parameters and the respective timestamps. The local database implemented at the Raspberry pi level is configured to be replicated at the remote server's side database, of which acts like a slave and copies the data from the master 1:1, every change done in the master is replicated by the slave. For the server side a set of PHP scripts were developed, the scrips being accessed by the mobile application that retrieve the sensors' stored data.

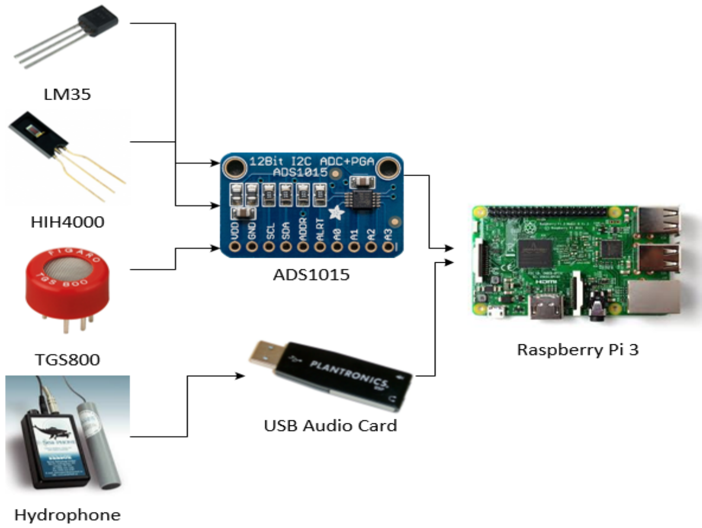

Fig. 1 - Hardware System Architecture

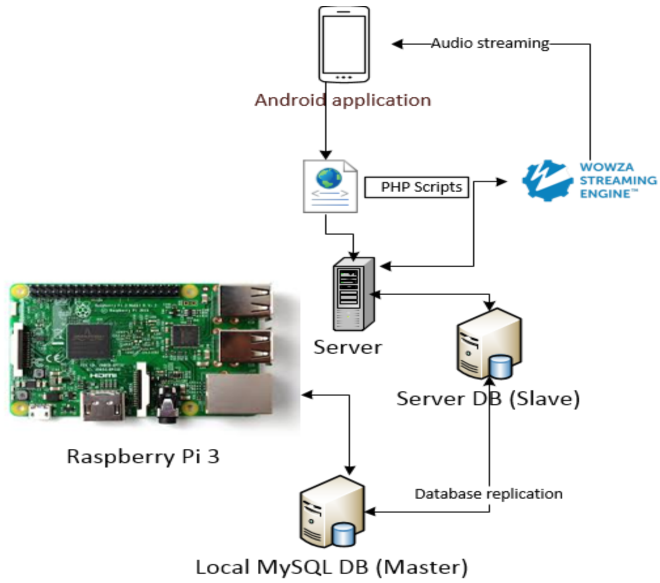

Fig. 2 - Software System Architecture

The audio streaming is handled by aWowzastreaming service [18] installed in the server. The service Wowza Streaming Engine facilitates the streaming from the Raspberry Pi to the server and from the server to the mobile application.

As for communication is concerned, a 3G/4G connection is used as it is more viable due to the distances with the drone, being easier to provide internet connection to the Raspberry Pi via a cellular receiver.

IV. Hardware and Connections

In this section, thedetails of the used hardwareare provided focussing in acquisition modules, sensors and processing units.

A. Acquisition modules

Since the Raspberry Pi does not includes analog inputs additional hardware is required to acquire the signals from analog sensors, An ADC module (Adafruit model) characterized by I2C communication interface was connected directly to the Raspberry through the GPIO ports with proper configuration. The ADC board is powered by 5V and has 4 analog input (A0-A4) that are used by the analog sensors.Using the python library provided by Adafruit the acquisition control is carried out. Using the developed scrips,the system startsgetting information from the sensors' channels.

The Plantronics USB audio adapter features a C-Media chipset which does not required any additional drivers presents full support for arm-based Linux distributions, making it ideal to work with a Raspberry Pi. It allows the acquisition the signals delivered by hydrophone measurement channel.

B. Sensing and conditioning circuits

The LM35 is a compact temperature sensor that can be powered by 5V; it provides an accuracy of 0.5ºC for the -55ºC to 150ºC specified measurement range. The sensor is already calibrated directly in Celsius and it provides an output voltage linearly proportional to the centigrade temperature [20].

The temperature sensor is powered with 5V from the Raspberry and the output is connected directly to the ADC module without additional conditioning required. The implemented voltage to temperature conversion is given by:

$$Temp(°C)=Vout\cdot 100 \quad (1)$$

where Vout is the output voltage of the LM35.

This humidity sensor provides voltage values that might be converted in relative humidity (RH) values [21]. The implemented relation to extract the RH values is

$$RH(\%) = \frac{\frac{Vout}{Vsupply} - 0.16}{0.0062} \quad (2)$$

where Vout is the output voltage of the sensor and Vsupply is 5V.

The TGS800 gas sensor from Figaro is recommended for air quality control measuring general air contaminants such as: carbon dioxide, ammonia, methane, ethanol and hydrogen through their gas concentrations in ppm. The sensor is characterized by a heater; the applied voltage VH on the heater is responsible for heating up the sensor making it sensitive to the pollution compounds. An additional circuit includes the sensor and a reference resistance. The circuit is powered by 5V and the output pin connects to the ADC board [22].

The ADC module presents a limitation due to being only 12-bit, both humidity and gas sensors easily saturated with the increase of output voltage which would reach the limit of 4.096V from the ADC. In order to solve this problem, a voltage divider was applied to attenuate the output voltage in half and then compensated back at a software level.

C. Computational Platform

The Raspberry Pi 3 is the core of the system; it deals with all the important data input and output. It receives data from all the analog sensors and the hydrophone and saves the data in a local database and replicates to a remote server and also streams the audio received by the hydrophone.

Several characteristics of Raspberry Pi are: possesses an ARM CPU, graphics capabilities, USB ports, Ethernet port,

HDMI port, programmable ports (GPIO) and a combined audio and composite video jack. Its characteristics made it ideal for this application. Thus it makes it ideal to use in the environment of this project, it is possible to connect all the sensors needed and to communicate with a server to keep uploading data. It also does not require much power (5V, ~1A), allowing it to be attached to a rather small battery for mobile purposes [23][24]. The operating system selected was the Raspbian, the official Raspberry Linux distribution, as it offers official support for specific Raspberry functionalities and not disregarding the advantages of having a Linux kernel.

V. SOFTWARE

The software component of the system includes embedded software and server software several details regarding backend and frontend functionality within the Raspberry Pi, databases, remote server and the mobile application are described.

A. Embedded platform software

Several scripts were described for the Raspberry Pi. The developed software allows the system to work flawlessly. The used software technologieswere Python, MySQL and FFmpeg.

A Python scripts were developed for data acquisition thus Raspberry is able to obtain the data from the measurement channels including sensors. The Adafruit's ADC library was used in this case.

The sensor data is uploaded to the local MySQL database installed in the Raspberry pi SDcard.. The script starts by executing a thread for each sensor with a user inputted reading interval; it then keeps reading the data from the ADC module every X seconds and uploading to the database at the same time. At this stage the script only ends with a SIGINT signal sent from the keyboard combination CTRL+C which terminate all the existing threads.

The MySQL database has a straightforward design, a table is created for each sensor and their fields are: id, value and timestamp. This allows the data storage associated with every sensor reading and the respective time of when the reading action was carried out.

To be able to stream the audio from the hydrophone, the FFmpeg software was installed to allow a streaming channel to be made to the server. The FFmpeg receives the audio from the ALSA input microphone channel, which is the general software in Linux to control all the audio/video devices and provides drivers in order for the devices to work in Linux. It then captures the microphone data and creates an RTMP channel that publishes the stream to the remote server.

B. Server side software

The server houses a couple of PHP scripts to allow the mobile application to access the sensors data. A total of 4 scripts were developed, three of them will return all of the saved data from each sensor in a JSON format and the last script will return the latest recorded value for each sensor, depending on the "GET" variable is passed by the mobile application. If a GET string "temp" is passed, the script returns the latest temperature value recorded the same goes for the "hum" and "gas" strings for their respective sensors. The purpose of this last script is to allow the mobile application to keep getting the latest value when it desires to show live readings with a certain interval.

In order of the streaming to work, it is also necessary to configure the server to act as the distributer of the audio stream. It receives a stream publication from the FFmpeg software in the Raspberry Pi and the mobile application will subscribe to that same stream via the same RTMP link. For all this to work, a streaming service from Wowza, the Wowza Streaming Engine was installed in the server to facilitate the streaming process. Wowza generates a link where it is allowed to publish streams with almost any audio/video streaming protocol and at the same time allows for subscriptions that lets a client software to play the audio/video, in this case, the mobile application.

C. Android Application

The android application is at this moment in an early stage, it has a very simple UI that shows the sensors latest reading values in Card Views. It reads the data through connecting to the remote server via HTTP and executing the PHP scripts that will return the data.

VI. Preliminary Results and Discussion

The implemented system was partially tested by components before installation on a USV. The sensing channels were tested one by one in the laboratory conditions.

For the temperature and relative measurement case the acquired and processed values are presented in the Fig. 3.

As expected, the values do not vary too much as it is an indoor measurement during the night, temperature was always around ~26ºC and humidity always around ~39%.

As for the gas concentrations, the following figure shows the sensor being turned on for a couple of hours and it can be seen that the sensor takes around 2-3 hours to calibrate.

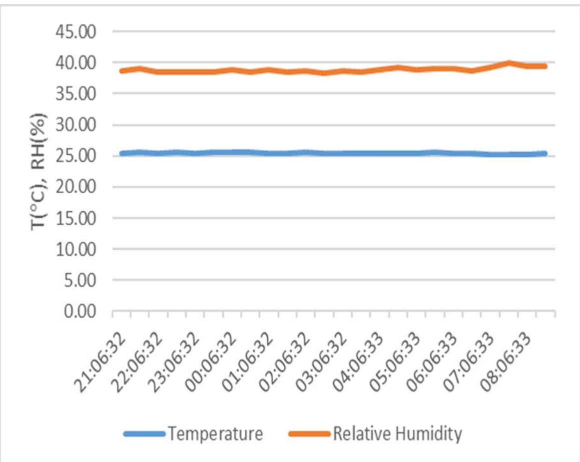

Fig. 3 – Temperature(T) and Relative Humidity (RH) Measurements in Indoors conditions

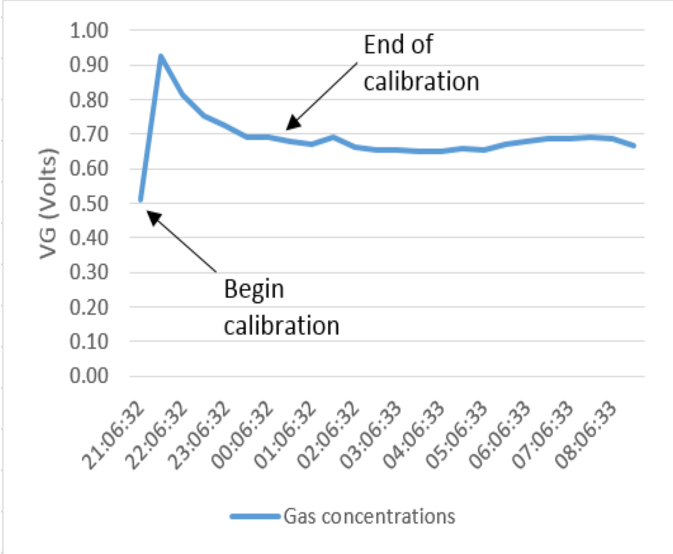

Fig. 4 – Gas Sensor Output Voltage (VG) evolution according with indoor gases' concentrations conditions

In this case the ethanol concentration was put really close to the sensor, explaining the high peak value. As the ethanol is removed from the air, the sensor takes time to adjust to normal levels as the rest of the gas inside the sensor evaporates.

Regarding the pollution event detection capabilities of the implemented system an ethanol pollution event was induced the broad band gas sensor response being presented in Fig. 5.

Several tests were also performed using the hydrophone mobile software component being designed and implemented to extract information about underwater audio spectral components. In Fig. 6 is presented the GUI of the underwater acoustic spectrum.

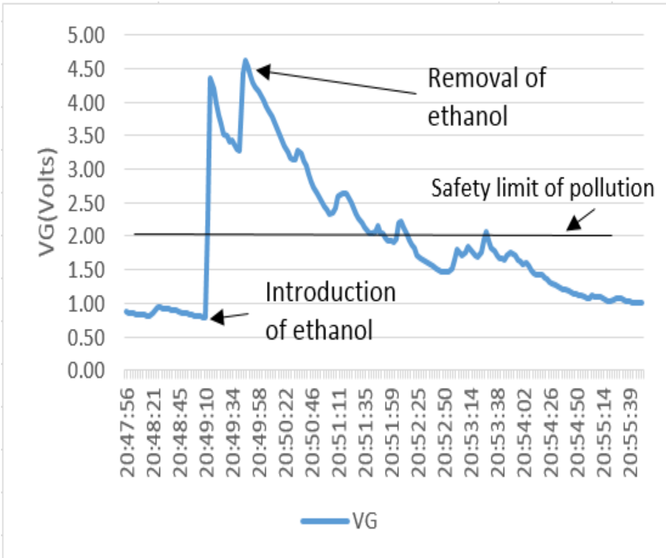

Fig. 5- Gas Sensor Output Voltage (VG) evolution with the presence of ethanol

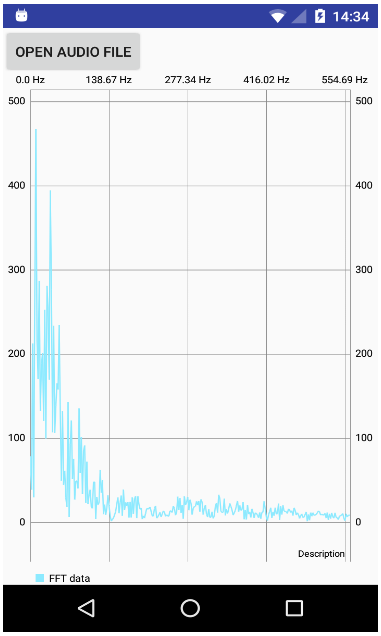

Fig. 6–Spectral analysis of audio signal provided by hydrophone implemented in the mobile APP

In the mobile application case, the GUI associated with on-line monitoring of air and water quality parameters is presented in Fig. 7. This is the main screen of the mobile application, where the measuredvalues delivered by sensors are displayed "in cards" and are updated in real-time. The capabilities of the implemented system on pollution event detection were also tested. Thus an ethanol pollution event was induced and the broad band gas sensor response is represented in Fig. 5.

The mobile application receives the data from the remote server via the implemented PHP scripts.Based on the remote server scripts the signal data after processing can be transferred to the Raspberry Pi.

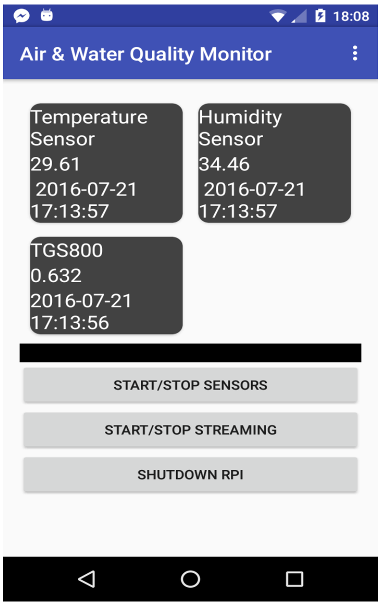

Fig. 7–Air & Water Quality Mobile APPs GUI

VII. Conclusions and Future Work

The goal of this paper is to provide information about UAV prototype implementation for water and air quality monitoring. The preliminary data underline the capabilities of the implemented system to provide information about water and air quality for an extended area. Future work will be focussed on mobile application test and improvement regarding functionalities and robustness. On the software side, the mobile application is also developed to provide data analysis and audio streaming. Additionally,field tests related to water and air quality monitoring together the implementation of measurement channel calibration procedures that will assure the appropriate measurement accuracy for the system will be carried out.

References

[1] O'Brien, H. Michael, 2016, The Internet of Things. *Journal of Internet Law*. 2016. Vol. 19, no. 12, p. 1-20.

[2] Bughin, Jacques, Chui, Michael and Manyika, James, 2015, An executive's guide to the Internet of Things. *Mckinsey Quarterly*. 2015. No. 4, p. 92-101.

[3] Wang, Biao, Lu, Shi-qiang, Lin, Wei-qing, Yang, Yi-fan and Wang, Dao-zeng, 2016, Water quality model with multiform of N/P transport and transformation in the Yangtze River Estuary. *Journal of Hydrodynamics*. 2016. Vol. 28, no. 3, p. 423-430.

[4] Velez, Fernando J., Nadziejko, Aleksandra, Christensen, Anders Lyhne, Oliveira, Sancho, Rodrigues, Tiago, Costa, Vasco, Duarte, Miguel, Silva, Fernando and Gomes, Jorge, 2015, Wireless Sensor and Networking Technologies for Swarms of Aquatic Surface Drones. *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*. 2015. DOI 10.1109/vtcfall.2015.7391193. Institute of Electrical & Electronics Engineers (IEEE)

[5] Zahugi, Emaad Mohamed H., Shanta, Mohamed M. and Prasad, T.V., 2012, Design of Multi-Robot System for Cleaning up Marine Oil Spill. *International Journal of Advanced Information Technology*. 2012. Vol. 2, no. 4.

[6] Burange, Anup W. and Misalkar, Harshal D., 2015, Review of Internet of Things in development of smart cities with data management & privacy. *2015 International Conference on Advances in Computer Engineering and Applications*. 2015. DOI 10.1109/icacea.2015.7164693. Institute of Electrical & Electronics Engineers (IEEE)

[7] Spalazzi, Luca, Taccari, Gilberto and Bernardini, Andrea, 2014, An Internet of Things ontology for earthquake emergency evaluation and response. *2014 International Conference on Collaboration Technologies and Systems (CTS)*. 2014. DOI 10.1109/cts.2014.6867619. Institute of Electrical & Electronics Engineers (IEEE)

[8] Ye, Jiuyan, Chen, Bin, Liu, Qingfeng and Fang, Yu, 2013, A precision agriculture management system based on Internet of Things and WebGIS. *2013 21st International Conference on Geoinformatics*. 2013. DOI 10.1109/geoinformatics.2013.6626173. Institute of Electrical & Electronics Engineers (IEEE)

[9] Duarte, Miguel, Costa, Vasco, Gomes, Jorge, Rodrigues, Tiago, Silva, Fernando, Oliveira, Sancho Moura and Christensen, Anders Lyhne, 2016, Evolution of Collective Behaviours for a Real Swarm of Aquatic Surface Robots. *PLOS ONE*. 2016. Vol. 11, no. 3, p. e0151834. DOI 10.1371/journal.pone.0151834. Public Library of Science (PLoS)

[10] Vijayakumar, N and Ramya, R, 2015, The real time monitoring of water quality in IoT environment. *2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]*. 2015. DOI 10.1109/iccpct.2015.7159459. Institute of Electrical & Electronics Engineers (IEEE)

[11] Waterontheweb.org, Water on the Web | Data |. [online]. [Accessed 13 July 2016]. Available from: http://www.waterontheweb.org/data/index.html

[12] DiNardo, P.B. and Andrews, S.B., [no date], Unmanned systems initiative: supporting immediate warfighter needs and army

transformation. *ROMAN 2005. IEEE International Workshop on Robot and Human Interactive Communication, 2005.*. DOI 10.1109/roman.2005.1513765. Institute of Electrical & Electronics Engineers (IEEE)

[13] Patterson, Mark R. and Patterson, Susan J., 2010, Unmanned systems: An emerging threat to waterside security: Bad robots are coming. *2010 International WaterSide Security Conference*. 2010. DOI 10.1109/wssc.2010.5730271. Institute of Electrical & Electronics Engineers (IEEE)

[14] Magnuson, Stew, 2016, Military beefs up research into swarming drones. *National Defense*. 2016. Vol. 100, no. 748, p. 22.

[15] Elston, Jack, Argrow, Brian, Stachura, Maciej, Weibel, Doug, Lawrence, Dale and Pope, David, 2015, Overview of Small Fixed-Wing Unmanned Aircraft for Meteorological Sampling. *Journal of Atmospheric & Oceanic Technology*. 2015. Vol. 32, no. 1, p. 97-115.

[16] Unmanned Aircraft Systems Overview, 2016. *Congressional Digest*, Vol. 95, no. 6, p. 2-32.

[17] Rao, Bharat, Gopi, Ashwin Goutham and Maione, Romana, 2016, The societal impact of commercial drones. *Technology in Society*. 2016. Vol. 45, p. 83-90. DOI 10.1016/j.techsoc.2016.02.009. Elsevier BV

[18] Media Server Software | Wowza Streaming Engine. *Wowza.com* [online]

[19] ADS1015 / ADS1115 | Raspberry Pi Analog to Digital Converters | Adafruit Learning System.*Learn.adafruit.com* [online]

[20] LM35 Precision Centigrade Temperature Sensors. *Ti.com* [online]

[21] HIH-4000 Series Humidity Sensors. *farnell.com* [online]

[22] Figaro Gas Sensors. *jens.files.wordpress.com* [online]

[23] Raspberry Pi, What is a Raspberry Pi? [online]. [Accessed 12 July 2016]. Available from: https://www.raspberrypi.org/help/what-is-a-raspberry-pi/

[24] Raspberrypi.org, Raspberry Pi Hardware - Raspberry Pi Documentation. [online]. [Accessed 12 July 2016]. Available from: https://www.raspberrypi.org/documentation/hardware/raspberrypi/README.md